10/531674

Rec'd PCT/PTO 27 MAR 2006

Phase locked loop and method for estimating the phase in a digital communication system

Technical field

The present invention relates to the field of digital communications, and in particular to a method for estimating the phase in a digital communication system and to a phase-locked loop.

Background of the invention

In a digital communication system comprising a transmitter and a receiver, a digital signal to be transmitted – usually a succession of symbols – is converted, before transmission, into a continuous-time analog signal, which is then transmitted through a physical propagation medium, air or any other physical propagation environment. When the signal is received by the receiver, it is then processed and converted into digital form by means of appropriate sampling, which is typically carried out at a frequency f_e that should be synchronous to symbol emission frequency f_s. Unfortunately, the clocks situated in the oscillating circuits equipping the transmitter and the receiver are never synchronous and it is then necessary to compensate for any frequency drift between these clocks, in order to be able to correctly process the received signal and to extract emitted symbols.

Such frequency shift affecting oscillators at transmission and reception generates a parasitic phase shift in the output signal of the complex demodulator located in the receiver. Other factors contribute to accentuate this parasitic phase shift. First, there is the time needed by digital signals to flow through a propagation medium. Secondly, any movement of the transmitter relative to the receiver generates Doppler beat and tends to introduce further disruptive phase shift.

Referring to a baseband model, observations Y_k at the output of the complex demodulator located in the receiver can be expressed by the following formula:

$$Y_k = a_k\, e^{i\xi_k} + \eta_k$$

where a_k corresponds to emitted symbols, ξ_k is the parasitic phase shift and η_K is additional noise.

Techniques are already known - based on phase estimator circuits - to estimate this parasitic phase shift ξ_K and correct it.

The most sophisticated phase estimators, that process simultaneously whole sets of received observations, are based on extremely cumbersome digital processing: particulate filtering, random-walk methods with Markov chain etc ... In practice, implementing such techniques proves to be unrealizable due to the considerable computing power required.

For this reason, the complexity of phase estimators is opposed to the simple implementation of phase-locked loops that sequentially process received observations one after the other instead of processing whole sets of received observations. Typically, a phase locked loop (PLL) is based on an iterative digital algorithm for estimating a phase estimate. Whereas traditionally, phase-locked loops were carried out by means of analog circuits, now such processing is purely digital. It should be noted that this digital processing depends closely on the type of modulation considered.

As an example, let us consider the case of a binary phase shift keying (BPSK) modulation. In such BPSK modulation, transmitted symbols a_k are equal to -1 or $+1$. Because of the previously mentioned parasitic phase shift, one doesn't obtain -1 nor $+1$ at the output of the complex demodulator, but these values modified by a phase shift. A well-known PLL for correcting such phase shift is the one known as Costa's loop that relies on the use of a gradient algorithm, associated with a cost function J given by the following formula:

$$J(\varphi) = E(\left|y^2_k e^{-i2\varphi} - 1\right|^2)$$

where E is the Expectation operator.

Applying the gradient algorithm to variable φ makes it possible to make the algorithm converge towards a phase estimate:

$$\varphi_k = \varphi_{k-1} - \gamma\, \partial J(\varphi) / \partial\varphi \Big|_{\varphi = \varphi_{k-1}}$$

A Costa's loop is finally obtained by removing the expectation: it is the stochastic gradient algorithm minimizing cost function J.

Other formulas are known for other types of modulation and in particular squaring modulation, also known as four-state Quadrature Amplitude Modulation or 4-QAM. Generally, whatever the type of modulation employed, phase-locked loops are built according to a general formula of the type:

$$\varphi_k = \varphi_{k-1} - \gamma F(Y_k, \varphi_{k-1})$$

where F is a function depending closely on the considered modulation.

Typically, as can be seen from the preceding formula, all loops consist in calculating a phase φ_k according to the preceding element φ_{k-1} and a function F of both elements Y_k and φ_{k-1} It should be noted that, in this formula, according to the type of phase shift to be corrected, sophisticated correction of the parameter γ can be used, and in particular a corrective second-order filter (proportional integral), or even a higher order filter, could be used.

All known phase-locked loops – conventionally adopting the known analog model - present the same limitation. The evaluation of phase φ_K is primarily based on the preceding phase value φ_{k-1} and on a function of one or more previous observations. Hence, an imperfect estimation of the phase and, consequently, correction thereof.

It is advisable to improve the phase locked loop (PLL) model in order to increase precision of the estimate and effectiveness of the correction process.

Disclosure of the invention

An object of the present invention is a new phase locked loop (PLL) structure making it possible to increase phase estimate precision compared to a conventional loop.

Another object of the invention is to provide a phase estimation process for a digital receiver, which is perfectly adapted for processing a digital signal in a receiver equipped with an error corrective system.

According to the invention, these objects are reached by means of a phase estimation method and a phase locked loop (PLL) device.

The invention provides a phase estimation method in a digital communication system comprising:

- receiving and storing a block of observations Y_K;
- executing at least one phase locked loop (PLL) on a predetermined sequence of observations extracted from said block.

The invention also provides a phase locked loop (PLL) device for a digital receiver including:

- means to receive and store blocks of observations;
- a first phase locked loop (PLL) for generating a first intermediate value;
- a second phase locked loop (PLL) for generating a second intermediate value;
- means to derive a phase estimate from said first and second intermediate values.

Using block processing at a phase locked loop (PLL) – whereas block processing usually takes place after the loop, at the level of error correcting codes – allows great improvement of the phase estimation process while preserving the very simple implementation of traditional phase-locked loops.

In particular, the chronological link between observations and the iterative algorithm implemented by PLL can be broken. The invention consists in associating two a priori paradoxical concepts, namely block processing and phase locked loop. In conventional approach, a phase loop is considered as an iterative processing, based on previous observations from which a phase estimate at a given moment is derived. The present invention goes beyond this approach and provides a process in which, after an observation block is stored, phase can be estimated from one or more PLLs, based on any sequence of observations within this block.

Preferably, a first phase locked loop (PLL) executing on a first set of observations of said block and a second phase locked loop (PLL) executing on another sequence of observations extracted from said block are realized. The results of both loops are then combined to provide a sharpened phase estimate at each sampling time.

Clearly, combining block processing with the simplicity of phase-locked loops allows great improvement of the phase estimation precision and, consequently, the phase correction. Indeed, with block processing previous observations as well as later observations within a block are taken into account for calculating φ_K. Hence a significant improvement of the phase locked loop (PLL) precision.

Although this process introduces a processing delay related to the storing of a whole block of observations before phase estimation for each observation Y_K, this delay is not prejudicial and is perfectly adapted to the use of some error correcting codes, like turbo-codes or block codes.

Preferably, block processing will be realized by means of a first phase locked loop operating in the chronological direction of the observations, and of a second phase locked loop operating in opposite direction.

In a preferred embodiment, the second phase locked loop (PLL) is initialized to the value provided by the first loop at the end of iteration.

Description of the drawings

Other features, objects and advantages of the invention will appear when reading the following description and drawings, only given by way of nonrestrictive examples, where:

Fig. 1 illustrates a flowchart of a phase estimation process according to the invention.

Fig. 2 illustrates the effect of a phase shift for a BPSK modulation.

Fig. 3 illustrates a preferred embodiment based on the use of two PLLs operating in opposite directions.

Figure 4 illustrates the application of the process of figure 1 to a 4-PSK or 4-QAM modulation.

Figure 5 illustrates the application of the invention to a two-state BPSK-type modulation.

Description of a preferred embodiment

Referring to Fig. 1, a general phase estimation process is described, which is appropriate for any type of modulation, BPSK, 4-QAM etc .... The process comprises a first step 11 where a set of n+1 observations (Y_0 to Y_n) output from a complex demodulator is received and stored within a block of n+1 observations. Typically, a block of about 1000 to 10 000 observations could be considered and people qualified in the art will choose such number according to the application and the type of modulation considered.

After storing the block, a first phase locked loop (PLL) is realized, which will execute on a predetermined sequence of block observations. Typically, any

sequence could be considered and more particularly a chronological sequence of observations Y_k, according to data reception order. Realizing a phase locked loop (PLL) is not particularly difficult and, for clarity's sake, will not be further developed. It is sufficient to say that, depending on the type of modulation used, an iterative algorithm according to the previously mentioned general formula is used.

$$\varphi_K = \varphi_{k-1} - \gamma F (Y_k , \varphi_{k-1}) \text{ with } k = 1 \text{ to } n$$

In step 12, a first phase locked loop is initialized. Typically, this initialization makes it possible to set the first values of the iterative algorithm, and in particular the first value φ_0. Preferably, when the phase is continuous from one block to the other, the first loop will be advantageously initialized by taking into account the last estimate from the preceding block. For a loop having an order higher than 1 several parameters will have to be set and, again, any useful continuity could advantageously be used.

In step 13, the first phase locked loop (PLL) is executed in order to build sequence φ_0, φ_1, φ_2, φ_3 .... φ_N, as illustrated in Fig. 3 by the arrow pointing right.

In step 14, a second phase locked loop (PLL) executing in opposite direction from the first PLL (as shown in Fig. 2 by right-left arrow). Preferably, the first value of the second loop, namely φ'_N, is initialized to the last digital value φ_N calculated by the first phase-locked loop.

In step 15, the second phase locked loop (PLL) is executed in order to build sequence φ'_{n-1}, φ'_{N2}, φ'_2, φ'_1 .... φ'_0 calculated in reverse direction compared to previously.

$$\varphi'_K = \varphi'_{k+1} - \gamma F (Y_k , \varphi'_{k+1}) \text{ with } k = n\text{-}1 \text{ to } 0$$

In step 16, intermediate results produced by the first and the second phase locked loops are combined in order to produce phase estimate φ''_k according to the following formula:

$$\varphi''_k = G(\varphi_k, \varphi'_k)$$

where function G is adapted to the type of modulation considered. In a particular embodiment, G is chosen so as to generate a weighted total in the form:

$$\varphi''_k = A \times \varphi_k + B \times \varphi'_k$$

Preferably, variable coefficients A_k and B_k could be chosen in order to give more importance to one of the phase-locked loops according to k. Indeed, the «weights» of the weighted total can be chosen in order to give more importance to the first loop in the right-hand part of the block of Fig. 2 and, conversely, to add more weight to the second loop in the left-hand part of the block. Thus, the loop having performed the most iterations is always given more importance and will reach a higher degree of accuracy in phase calculation.

Referring to Fig. 4, application of the general process of Fig. 1 to a 4-BPSK or 4-QAM-type modulation is now described. In this type of modulation, complex symbols a_k, which are respectively –1, +1, -i and +i, are transmitted. The process begins at step 41 that consists in receiving and storing a block of observations received from the complex demodulator, like previously.

In step 42, the first phase locked loop (PLL) adapted for a 4-QAM modulation is initialized:

$$\varphi_K = \varphi_{K-1} + \gamma \operatorname{Img}(y^4_K e^{-i4\varphi(k-1)})$$

where Img corresponds to the imaginary part of the complex number.

It should be noted that, in practice, using a sophisticated digital filter could be considered for calculating factor γ according to the phase shift model to be corrected. For simple phase shifts, a simple proportional corrector could be enough whereas in more complex cases, it would be advantageous to use an integral corrector, or even a high-order filter.

For phase locked loop initialization, the first value φ_0 is given. Preferably, filter γ will be realized by means of a second-order digital filter, for example a second-order filter according to the following formula:

$$\gamma = \gamma_1 + \gamma_2 / (1 + z^{-1})$$

and it will be initialized while taking into account all usable continuity factors.

In step 43, the first phase locked loop (PLL) is executed in order to build sequence $\varphi_0, \varphi_1, \varphi_2, \varphi_3 \ldots. \varphi_N$.

In step 44, a second Costa's PLL is now initialized to the last value calculated by the first loop and, in step 45 said second loop is executed to build the sequence $\varphi'_{n-1}, \varphi'_{N2}, \varphi'_2, \varphi'_1 \ldots. \varphi'_0$ calculated in the opposite direction.

$$\varphi'_K = \varphi'_{k+1} + \gamma \mathrm{Img}\,(y^4_k\, e^{-i4\varphi\,(k+1)}) \qquad k=n-1 \ldots 0$$

These intermediate results are then combined at step 46 to generate the phase shift estimate φ''_K, preferably according to the formula:

$$\varphi''_K = G\,(\varphi_K, \varphi'_K)$$

In step 47, processing of the current block is completed.

To illustrate the general nature of the invention, with reference to Fig. 5, we will now show how the invention can contribute to considerably improve phase estimation in a BPSK modulation.

Again, the process starts with the reception and storing a block of observations Y_k, step 51.

In step 52, the first phase locked loop (PLL) adapted for a BPSK modulation is initialized. In a new and particularly advantageous way, the first PLL is combined with the use of statistical data related to symbols a_K transmitted sin this BPSK

modulation. To this end, a phase locked loop (PLL) according to the following formula is realized:

$$\varphi_K = \varphi_{k-1} + \gamma \operatorname{Img} (y_k e^{-i\varphi (k-1)} \operatorname{th}[L_k / 2 + 2 / \sigma^2 \operatorname{Re}(y_k e^{-i\varphi (k-1)}])$$

where:

th is the hyperbolic tangent operator,
Re is the operator of the real part of a complex number,
σ^2 is the noise variance;
and $L_k = \operatorname{Ln} [p(a_k =1) / p(a_k = -1)]$,

and Ln is the natural logarithm, p(ak =1) is the probability that symbol a_k is equal to +1 and $p(a_k =-1)$ is the probability that symbol a_k is equal to -1.

As previously, it would be possible to use a second-order digital filter – even a higher order filter – for factor γ.

In step 53, the first phase locked loop (PLL) is executed in order to build sequence $\varphi_0, \varphi_1, \varphi_2, \varphi_3 \ldots \varphi_n$.

In step 54, a second phase locked loop (PLL) built like previously is now initialized to the last value calculated by the first loop and, while taking into account statistical data related to symbols. In step 55 this loop is executed to build sequence $\varphi'_{n-1}, \varphi'_{n-2}, \varphi'_2, \varphi'_1 \ldots \varphi'_0$ calculated in the opposite direction as compared to previously.

Two alternative embodiments are still possible. If noise power stays low, hyperbolic tangent function can be approximated using signum function. The following formula is then obtained, which is typical of a decision feedback loop, to within term $L_k/2$.

$$\varphi_k = \varphi_{k+1} + \gamma \operatorname{Img} (y_k e^{-i\varphi (k+1)} \operatorname{th}[L_k / 2 + 2 / \sigma^2 \operatorname{Re}(y_k e^{-i\varphi (k+1)})])$$

On the other hand when noise power is high, it can be noted that the hyperbolic operator tangent can be approximated with an identity function, which comes back to a Costa's loop formula (to within addend L_k / 2).

In step 56, intermediate results obtained from both phase-locked loops are combined to generate the phase shift estimate φ''_K, preferably according to the formula:

$$\varphi''_K = G(\varphi_K, \varphi'_K)$$

As previously, processing of the current block is completed in step 57.

Both preceding examples show that the process according to the invention applies to any type of modulation and any type of phase locked loop (PLL). Clearly, people qualified in the art could readily realize all appropriate adaptations, and even advantageously combine a very early phase locked loop block processing with any subsequent later processing, like error correcting codes, turbo-codes etc ...